

## From Declarative Knowledge to Process-based Crisis Resolution: Application to Flood Management

Hanane Ariouat  
IRIT/UT1-Capitole  
[Hanane.Ariouat@ut-capitole.fr](mailto:Hanane.Ariouat@ut-capitole.fr)

Eric Andonoff  
IRIT/UT1-Capitole  
[Eric.Andonoff@ut-capitole.fr](mailto:Eric.Andonoff@ut-capitole.fr)

Chihab Hanachi  
IRIT/UT1-Capitole  
[Chihab.Hanachi@ut-capitole.fr](mailto:Chihab.Hanachi@ut-capitole.fr)

### Abstract

*Crisis resolution is often based on official government plans that provide guidelines. In real time, when a crisis occurs, one or several plans have to be chosen, merged, refined to meet the specific requirements of the crisis, and then launched. Plans are often in a textual format, which makes their interpretation ambiguous and error prone. Therefore, in real time, the coordination of stakeholders becomes difficult and time consuming. Given these drawbacks, the transformation of a plan into a process provides several advantages: i) an accurate and machine-readable specification of coordination of actions to be done in the field, ii) a better common understanding between stakeholders responsible for these actions and iii) a mean to analyze, simulate and evaluate the crisis response before launching it. The problem being addressed in this paper is “how to deduce a process for driving crisis resolution from business knowledge (plans, stakeholders and their capacities) and relevant facts observed in the impacted field”. This paper presents first a meta-model for capturing business knowledge and crisis situation and then a deduction approach deriving a process in a BPMN-like format. Flood of the Loire in June 2016 serves as a support for approach experiment.*

### 1. Introduction

In crisis management, several participating actors (stakeholders) have to act simultaneously and urgently to reduce the crisis and its impacts on the real world [1]. To achieve this common goal efficiently, these actors must collaborate, or at least act in a coordinated way in order to make their activities as efficient as possible. In France, in civilian crisis management such as flood, fire, chemical accident or terrorist attack, this coordination definition is under the responsibility of a control center, called crisis cell, headed by a *Préfet* or by the *Ministre de l'Intérieur*, depending on the crisis

scale. The cell, which is composed of the representatives of different public organizations involved in crisis resolution and which may be geographically distributed, is in charge of applying governmental plans defined by the law and providing guidelines for action in response to the crisis [2]. However, these plans provide general guidelines and have to be adapted to meet crisis requirements. In addition, the coordination between actors involved in crisis resolution is not specified and must be defined by the crisis cell.

This paper deals with the design of the coordination of actions of actors involved in the field to reduce the crisis. It recommends a process-based approach to address this issue. Indeed, there is an easy mapping between governmental plans and processes [3]: actions and actors from governmental plans correspond to process activities and roles while coordination of actions is explicitly modelled in the process using coordination patterns such as sequence, alternative, or parallelism between activities [4]. Moreover, a process-based approach provides an understandable, accurate and machine-readable specification of actions to be done in the field as well as a means for analyzing, simulating and evaluating the crisis response before launching it.

More precisely, the paper contributes to make the design of this coordination easier for crisis cells. It is based on the following cycle driving crisis resolution. This cycle, visualized in Fig. 1, is made of three steps.

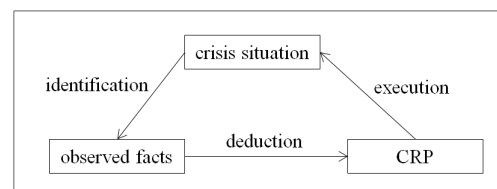


Fig. 1. Crisis resolution cycle

The first step is the identification of relevant facts from data recorded in the field. The second step is the deduction of the corresponding Crisis Resolution Process (CRP) taking into account business knowledge of actors involved in crisis resolution. Finally the third

step is the execution of relevant actions according to the CRP, which leads to the modification of the crisis situation.

The paper focuses on the deduction step, which is a fundamental step in the crisis resolution cycle. Deduction of CRPs has already been addressed in literature (*e.g.*, [5, 6]), but the main drawback of existing contributions is that they only support deduction of services organized in sequence or in parallel. They are unable to deduce choices and thus they are unable to model alternative plans that crisis cells have to consider. In addition these solutions fail in explaining why services are selected, both in terms of facts observed (a service is selected to deal with a risk or damage), and in terms of services (a service is selected as it is necessary to another one).

The current paper addresses these two previous issues by providing two main contributions: a meta-model including knowledge to derive and justify the CRP deduction and an algorithm able to mine a CRP including alternatives plans. Also an experimentation of our approach on a real case study, namely the flood of the Loire in June 2016 is given.

In the reminder of this paper, we first place our CRP deduction approach with regard to related work. Sec. 3 presents our meta-model defining the concepts needed to represent both facts describing a given crisis and business knowledge related to crisis resolution. Sec. 4 is dedicated to the deduction of CRP. It presents the deduction approach and introduces the main algorithms implementing it. Sec. 5 details our experimentation through a specific instantiation of our meta-model. Finally the conclusion summarizes the paper and mentions some open issues.

## 2. Related work

We have found several contributions advocating a process-based approach in the field of crisis management and each of them has implemented algorithms for process deduction (*e.g.*, [5, 6]). In these contributions, as in ours, the inferred process is expressed as a BPMN-like diagram. However, in each of these contributions, deduction differs from ours.

First the contribution described in [5] recommends a composition-based deduction from input and output of services that have to be deployed in the field. Both sequencing and parallelization of services can be deduced. More precisely a service having outputs corresponding to inputs of another one is represented by two BPMN tasks, corresponding to the considered services, linked together by a sequence flow. For instance, if all the outputs of service a correspond to all the inputs of service b, then services a and b are

modelled as tasks connected by a sequence flow from a to b in the BPMN diagram. In addition, when a service has outputs corresponding to inputs of several services, these last services are modelled as parallel services. For instance, if outputs of service a (*e.g.*, e1 and e2) correspond to all the inputs of service b and if the other outputs of service a (*e.g.*, e3) correspond to all the inputs of service c, then services b and c are modelled as parallel tasks executed after service a.

Second, the contribution described in [6] recommends a knowledge-based deduction. More precisely this work recommends a specific OWL ontology for both crisis situation and crisis response modelling. It also introduces specific SWRL rules for both selection of services to be deployed in the field and their ordering as sequencing or parallelization. This second contribution is interesting because, unlike the first one ([5]), which recommends a composition-based deduction according to inputs and outputs of services as in web service composition, it takes into account knowledge of actors involved in the field introducing the notion of objective and linking objectives with services to be deployed to reach them. However, this knowledge is expressed within rules which are difficult to define and hard-coded in the ontology. In addition, these rules only deduce sequencing and parallelization of services: they are unable to deduce alternatives to reduce risk or deal with damage. However, deducing alternatives for driving crisis resolution is really relevant for crisis cells, which need to know the possible solutions before making decisions.

We also have found several contributions addressing process deduction in the field of process mining [7]. The main one is the Alpha algorithm [8], which identifies process schemas from log file repositories. These log files record executed activities (services), actors performing them and corresponding timestamp. Alpha mines these log files to identify the coordination of executed activities by actors. Moreover, some works have addressed the deduction of activities to deal with a given situation. First, the contribution described in [9] recommends a deduction service that is implemented in the ProM tool and that aims at providing the user with the next possible activity to perform in a given situation. This service exploits process log repositories to match the running process with existing cases. In [10], the recommended system compares different process mining algorithms to identify the one which best fits a given situation. Comparison is based on measurements such as fitness and generalization, which allow the evaluation of the performance and the quality of the compared algorithms. Finally, [11] describes a recommender task system that uses social tagging to collect relevant

information from discussions between process actors during process execution. Analysis of these tags allows the system for deducing new tasks when the same process must be executed again.

The approach recommended in this paper is in line with the one described in [6]. As in [6], we advocate a knowledge-based deduction considering both facts observed in the field and business knowledge of crisis actors. More precisely business knowledge corresponds to already existing solution modelled as services offered by crisis actors and possible corresponding plans defining coordination between them. In addition, we also model relations of these services in terms of what do they require, what do they cause, why using one or another. These relations correspond to knowledge of crisis actors indicating how services must be performed in the field. Unlike [6], we do not model this knowledge as a set of hard-coded rules in an ontology but rather as data stored in a database. Thus our approach is declarative, making knowledge management easier. Moreover, we fully exploit this knowledge as we deduce choices, in addition to sequencing and parallelization of services. On the other hand, we are able to explain why services participate in the CRP as they deal with risk or damage, or as they are dependent from other participating services.

Moreover, we also exploit the powerful conventional process mining algorithms, notably Alpha [8], which extract process schema from file log repositories. Such algorithms are also helpful in crisis management for deducing CRP. However our CRP deduction approach differs from the process mining one as the activities (services) and their coordination are no more extracted from log files but rather deduced by a matching process and by using pre-existing relations between the services modeled in a crisis meta-model. Therefore, we start from scratch i.e. without execution cases (log files) and we guarantee to express all the possible scenarios. Finally, unlike [9–11], we focus on process deduction to deal with a situation as it is the essential requirement of crisis cells.

### 3. The crisis meta model

This section presents a meta-model for defining both crisis description (facts) and the required knowledge for crisis treatment (business knowledge), which are two fundamental dimensions that must be taken into account in crisis representation [12]. This resulting meta-model is given in Fig. 2 as an UML class diagram. While facts are case dependent, knowledge are specific to a crisis domain (flood, forest fire, earthquake...). They enable a declarative

description of the resolution process since its control structure, i.e. the coordination of the actions, is not explicitly described but derived/discovered from the facts and binary relations between actions. This aspect differentiates our proposition from other meta-models ([13]) where the process is explicitly described and requires a heavy work from the users. Also our meta-model includes both the concept of plan and services. A risk/damage could be associated to a set of individual services and/or to a resolution plan encapsulating a set of coherent services. This offers to the user flexibility to describe and organize in a modular way his knowledge.

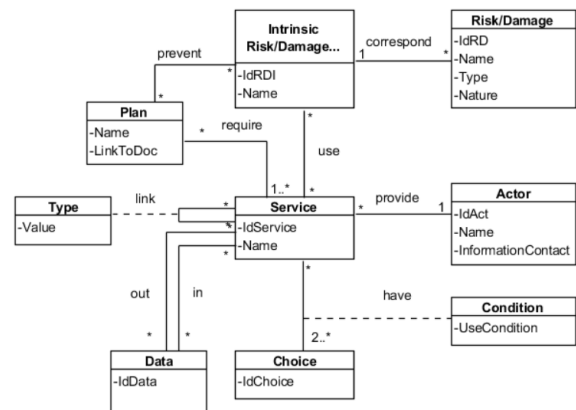


Fig. 2. Crisis meta-model

#### 3.1. Facts representation

The facts are abstracted in the *Risk/Damage class*. This concept corresponds to an observed fact in the field, which can either be risk or damage. Damage is a negative situation affecting for instance population (e.g., flooded house with people inside), building (e.g., flooded school), road (e.g., cut-off road)..., while risk is the potential for damage. For each risk or damage, we store its type (risk or damage), and its nature, i.e. if it is already known or not. When it is known, it is linked to the knowledge base and more particularly to the corresponding *Intrinsic Risk/Damage* (relationship *correspond*), which gives access to the known solution for treating it. When it is unknown, the crisis cell has to specify at run time how to deal with this new risk or damage, indicating which services to be deployed in the field (relationship *deal with*). In addition, for each risk or damage, we store a specific property indicating if the risk or damage has priority or not. A priority risk or damage has to be considered in the deduction process when deducing the CRP, while a not priority risk or damage will not be taken into account by the current deduction; it will be taken into account later, when another deduction is made. Crisis cell members

may change the value of this property according to the urgency of risk or damage.

### 3.2. Business knowledge representation

Business knowledge completes the facts with the following concepts: Intrinsic Risk/Damage, Plan, Service, Actor, Data, Choice, Condition and Type. Operational actions that can be executed in the field are modelled as services provided by crisis actors. For each service, we store its input and output data. Moreover, the rules relative to the use of these services are expressed as relations between services (relationship link). A relation between two services may be require, cause, or follow. Types require and cause define a strong relation among considered services, indicating that both services have to be executed one after the other: require indicates there is a precedence relation among them while cause indicates that there is a succession relation among them. At the opposite, the type follow defines a weak relation among considered services, indicating that a service will be performed after another, but not necessarily right after. In addition, we also have introduced another relation between services, namely the choice relation. The idea is to support alternative modelling, each alternative being a possible solution to deal with an issue. A condition defines when using this alternative. Finally services provided by actors may be used to deal with intrinsic risk, possibly as part of a plan, which corresponds to already specified set of actions to be undertaken to address an issue [1].

## 4. Deduction of crisis resolution processes

A Crisis Resolution Process (CRP) is a process driving crisis resolution. It includes the set of ordered actions (services) to be undertaken by crisis actors and their coordination.

### 4.1. Deduction principle

Basis of deduction are both facts observed in the field, which correspond to risks to be reduced or

damages to be repaired, and business knowledge of actors, specified as services, and relations between these services. Deduction principle is given in Fig. 3 as a BPMN process diagram. This process includes three main steps.

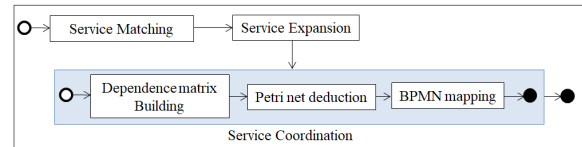


Fig. 3. Deduction Principle process

The first step is the *Service Matching* step, which matches observed facts with business knowledge and more precisely, intrinsic risk and damage and corresponding services. The result of this step is a minimal set of services to be deployed. This set is then completed in the *Service Expansion* step. To do this, we exploit the relations between services to identify additional services to be deployed. The result of this expansion step is the set of services to be coordinated in the corresponding CRP. Finally, the *Service Ordering* step is responsible for ordering services w.r.t. their relation. It is visualized as a sub-process in Fig. 3. First, we build a matrix describing dependences existing between considered services from relations existing between them. As in conventional process mining algorithms [7], we consider three types of dependences:

- causal dependence: a causal dependence between services  $a$  and  $b$ , denoted  $a \rightarrow b$ , indicates that service  $a$  has to be executed just before service  $b$ ,
- parallel dependence: a parallel dependence between services  $a$  and  $b$ , denoted  $a \parallel b$ , indicates that services  $a$  and  $b$  are executed in any order,
- unrelated dependence: an unrelated dependence between services  $a$  and  $b$ , denoted  $a \# b$ , indicates that services  $a$  and  $b$  are completely independent one from another, that is it does not exist any causal or parallel dependence between them.

Then, from this dependence matrix, we build the corresponding Petri Net from which we derive the corresponding BPMN-like diagram. The Petri net serves as a support for CRP analysis, simulation and

Algorithm ServiceExpansion (in out s: Set(Service)) # expands s with useful additional services Variable x: Service, tobeExpanded, Expanded: set(Service) Begin 1:   tobeExpanded = s 2:   Expanded = {} 3:   # Expanded is the resulting set of services built from tobeExpanded 4:   While tobeExpanded $\neq$ {} Loop 5:     # x is a service of tobeExpanded which does not belong to Expanded 6:     x = Select(tobeExpanded, Expanded) 7:     # tobeExpanded is completed with services in relation with x 8:     tobeExpanded = tobeExpanded / {x} $\cup$ require(x) $\cup$ cause(x) $\cup$ choice(x) 9:     # x is added to Expanded 10:    Expanded = Expanded $\cup$ {x} 11:   End Loop 12:   s = Expanded End
---

Algo. 1. algorithm ServiceExpansion

validation, while the BPMN serves as a support for CRP execution. Note that the Petri net formalism has been chosen as it provides formal and executable specifications to analyze, simulate, check and validate the described process [14] while BPMN has been chosen as it is the language of the process engine that we use in this work.

## 4.2. Algorithms for CRP deduction

Different algorithms have been written to implement CRP deduction. We present below the two main ones.

The first algorithm is the algorithm *ServiceExpansion*, detailed below in Algo. 1. This algorithm implements service expansion using choice, cause and require relations as follows. The idea is to add services that are required to, consequence of, or alternative to each service obtained after matching. For that we use two sets of services, namely *tobeExpanded*, whose initial value is the set of services obtained after matching, and *Expanded*, and we add to *Expanded* both a service *x* from *tobeExpanded* and services connected to *x* by require, cause or choice relation.

The second algorithm is the algorithm *ServiceCoordination*, detailed in Algo. 2. This algorithm deduces the CRP to be deployed in the field using a process mining-based approach. More precisely this algorithm extends the Alpha algorithm [8], which is the key algorithm in process mining, and which needs to be revisited to address the deduction of CRP service coordination. Unlike Alpha [8] in which the services (activities) and their coordination are mined from log file repositories, we deduce services in line with facts observed in the field (cf. Algo 1) and define their coordination by using pre-existing relations

between deduced services (cf. Algo 2). Therefore, we start from scratch, i.e. without examples of execution (log files) and we guarantee to express all the possible scenarios. While in Alpha a dependency between two activities is deduced from their direct succession in traces recorded in log files, we generate it from pre-existing relations between services. We also add artificial services to represent complex patterns such processes starting with parallel activities. Moreover, as in Alpha, the Petri net formalism supports process description in terms of places, transitions, corresponding to actions to be executed, and arcs, connecting places and transitions. However, as defended before, our algorithm also provides a BPMN-like representation of the CRP, which is more convenient to crisis cell members.

More precisely, our algorithm first builds the dependence matrix basis of the CRP deduction, but in a very different way from Alpha as we exploit business knowledge and not execution logs. To get into detail of this matrix building, for the set of services obtained after service expansion, we build causal dependences in the matrix from require, cause and follow relations (line #1 in algo 2). We also analyze relation between these services to eventually define new services which correspond to choices and specify unrelated dependences according to choice relations (line #2). Finally, parallel dependences between these services are deduced using the following rules (line #3):

If  $a \rightarrow b$  and  $a \rightarrow c$  and not ( $b \# c$ ) Then  $b \parallel c$   
 If  $a \parallel b$  and  $a \rightarrow c$  and not( $b \rightarrow c$ ) then  $b \parallel c$

Then, the construction of the Petri net is fairly similar to Alpha. However, even in this similar part, we extend Alpha adding specific places and transitions

<p>Algorithm ServiceCoordination (in s: {Service}, out pn: PetriNet_Diagram, out bpmn: BPMN_Diagram)</p> <p># deduces pn and bpmn from the set of services s</p> <p>Variable</p> <p>m: DependenceMatrix, T: Set(Transition), P: Set(Place), A: Set(Arc)</p> <p>Begin</p> <p>1: DetermineCausalDependence(m, s) # determine causal dependences between services in s</p> <p>2: DetermineUnrelatedDependence(m, s) # determine unrelated dependences between services in s</p> <p>3: DetermineParallelDependence(m, s) # determine parallel dependences between services in s</p> <p>4: # T is the set of services of m completed with two additional services to start and end the process</p> <p>5: <math>T_s = \{\text{Start}\}</math></p> <p>6: <math>T_e = \{\text{End}\}</math></p> <p>7: <math>T = \text{DetermineServices}(m) \cup T_s \cup T_e</math> # tasks of the Petri net</p> <p>8: # determine tasks t such as t is not part of right-hand side of any dependence</p> <p>9: <math>S_s = \text{DetermineNotRightHandSide}(m)</math></p> <p>10: # determine tasks t such as t is not part of left-hand side of any dependence</p> <p>11: <math>S_e = \text{DetermineNotLeftHandSide}(m)</math></p> <p>12: # build the Petri net</p> <p>13: <math>X = \{(A,B) / A \subseteq T \wedge B \subseteq T \wedge \forall a \in A \forall b \in B, a \rightarrow b \wedge \forall a_1, a_2 \in A, a_1 \# a_2 \wedge \forall b_1, b_2 \in B, b_1 \# b_2\}</math></p> <p>14: <math>Y = \{(A,B) \in X / \forall (A',B') \in X, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B')\}</math></p> <p>15: <math>Z = Y \cup \{(Ts,W) / W \subseteq T \wedge (\exists V \subseteq T / (V,W) \in Y) \wedge (\forall V' \subseteq T (V',W) \notin Y) \wedge (\forall w \in W / w \in S_s)\} \cup \{(W,Te) / W \subseteq T \wedge (\exists V \subseteq T / (V,W) \in Y) \wedge (\forall V' \subseteq T (W,V') \notin Y) \wedge (\forall w \in W / w \in S_e)\}</math></p> <p>16: <math>P = \{P_{(A,B)} / (A,B) \in Z\} \cup \{P_{(\emptyset, Ts)}, P_{(Te, \emptyset)}\}</math> # places of the Petri net</p> <p>17: <math>A = \{(a, P_{(A,B)}) / (A,B) \in Z \wedge a \in A\} \cup \{(P_{(A,B)}, b) / (A,B) \in Z \wedge b \in B\} \cup \{P_{(\emptyset, Ts)}, \text{Start}\} \cup \{\text{End}, P_{(Te, \emptyset)}\}</math></p> <p>18: <math>pn = (T,P,A)</math> # the resulting Petri Net pn is the triplet (T,P,A),</p> <p>19: <math>bpmn = \text{Map}(pn)</math> # which is then mapped in BPMN</p> <p>End</p>
--

Algo. 2. algorithm ServiceCoordination (adaptation of Alpha algorithm)

to build processes possibly starting with parallelism or alternative. More precisely, as Alpha, we identify initial and final services, which are services to be executed respectively at the beginning and at the end of the CRP. Then, the novelty is to define two virtual transitions: Start and End. Start is connected to each initial service so that they could be performed after Start. Also, each final service is connected to the End transition, so that the End transition merges the results of the final services. Another important difference with Alpha is that we are able to deduce alternatives involving empty activities as we automatically add these activities when building the matrix. Thus we overcome some limitations of Alpha (e.g., [15, 16]). Finally, the part of the algorithm inspired by Alpha is (i) the determination of X, the minimum set of couples (Servicesa, Servicesb) for which, each sa in Servicesa has a causal dependence with each sb in Servicesb as well as sa and sb are unrelated (line #13), (ii) the determination of Y, which is a subset of X (line #14) and (iii) the aggregation of the final Petri net (line #18). All other algorithm lines are specific to our CRP deduction.

The resulting Petri net is then mapped into a BPMN-like diagram (line #19), which does not include BPMN pool and lanes to be more readable for crisis cell members. We do not detail this mapping as it is quite classic (e.g., plug-in PROM supports mapping to BPMN from Petri net [17]), but we highlight its

specificities in GéNéPi. Indeed, in GéNéPi, BPMN is not only a notation for CRP visualization but also the executable process language of Iterop, the process engine that supports CRP execution. Thus to obtain a fully executable specification, we have mapped flowing conditions, i.e. conditions attached to sequence flow flowing from open exclusive gateways to activities (i.e., services) in the BPMN-like diagram. More precisely, if use conditions of services are defined in the meta-model, then these use conditions are the flowing conditions. Otherwise, the algorithm automatically adds an out data to the activity preceding an open exclusive gateway, and defines for each sequence flow flowing from this open exclusive gateway a condition in which this out data is involved. Another interesting aspect in this mapping is the labelling of services with the facts they deal with. Thereby the algorithm labels each service with the facts justifying the selection of the service in the CRP, making it possible to determine whether or not all activities related to a fact are carried out or not. Thus it is possible to modify crisis situation deleting facts processed from the list of facts to be taken into account. Finally, we simplify the CRP in removing Start and End services, which were introduced for consistency reasons when building the Petri net, but which are no more useful in the BPMN. We also remove added services in the Petri net for syntactic reasons but useless in the BPMN.

## 5. Approach experiment

We have conducted an experiment considering the flood of the Loire in June 2016 as a case study. This experiment has been set up in collaboration with the crisis cell of Orléans, prefecture of department 45 in France, in the context of the GÉNÉPi project<sup>1</sup>, which aims at making civilian crisis (notably floods) management easier for crisis cells. Orléans being often deeply affected by Loire's floods, their mastering is of utmost importance. Members of the crisis cell were the Préfet, head of prefecture, the COD, which is the operational committee set up within the crisis cell and finally the representatives of the different actors acting in the field (*e.g.*, DDT that are responsible for dykes supervision...). The experiment has focused on the simulation of several days of the last flood of the Loire in June 2016. This section introduces the considered case study and reports on the crisis meta-model instantiation and the CRP deduction.

### 5.1. Case study

Flood of the Loire in June 2016 lasted 12 days. In our experiment, in accordance with the crisis cell, we only focused on days 1, 3, 7 and 9, thus highlighting different interesting aspects of the response (*i.e.*, the deduced CRP) recommended by our deduction algorithm. This paper only reports on a simplified version of day 7, is complex enough to illustrate the value added by our algorithm to deduce complex processes that include crisis cell decision making and hierarchical communication towards different ministries to which crisis cell is accountable.

On day 7, rainfall forecast is substantial. Loire level should rise significantly and major concerns of crisis cell are the following facts:

- risk of civilian casualties in nursing home Saint Pryvé Lake: the nursing home has to be evacuated,
- risk of flooding of motorway A71: the motorway has to be partly cut off,
- risk of dyke failure in Saint Pryvé Saint Mesmin: municipality of Saint Pryvé Saint Mesmin, next to Orléans, could be flooded and some districts of the municipality could be evacuated.

### 5.2. Day 7: facts and knowledge modelling

The three previous risks are modelled as Risk/Damage in the recommended meta-model. Their priority is 1, they are known and are linked to three Intrinsic Risk/Damage via the relationship correspond.

These intrinsic risks serve as a basis for defining the required knowledge to address dyke failures, flooding of nursing homes and flooding of motorways. We do not detail below the modelling of risks but we rather focus on the modelling of services (without detailing their in and out data) along with their relation, as they serve as a basis for matrix dependence building.

On Day 7, 17 services, offered by 6 actors, have to be selected to participate into the CRP. These services and the corresponding actors are given in Table 1. In addition, Table 2 shows existing relations between these services and Table 3 gives their use conditions. On day 7, we only have two use conditions respectively for services Decision-making for evacuation and Dyke supervision continuation. Both conditions involve two out data from service Dyke state evaluation, namely *dykeState*, whose value is low or solid, and *dykeSupervision* whose value is yes or no.

### 5.3. Day 7: deduced CRP

As explained before we deduce a CRP as a Petri net diagram and we map it into a BPMN one. Due to lack of space, we only provide the BPMN-like diagram in Fig. 4.

Both diagrams are built from the dependence matrix given in Table 4. This dependence matrix is built from knowledge stored in the meta-model (*cf.* Tables 1, 2 and 3). More precisely, dependence matrix building exploits relations between services (require, cause, follow and choice). In this building, we also identify services that are useful for modelling choices. For instance, services 18 and 19 have been added automatically as we have use conditions for services 5 and 7 and no choice relation for each of them. As a consequence, the algorithm identifies choice relations between services 5 and 18 and between services 7 and 19 and produces the corresponding unrelated dependences. In addition, unrelated dependence is symmetric: both  $a\#b$  and  $b\#a$  are indicated in the matrix (*e.g.*, 5#18 and 18#5). Moreover, causal dependences are indicated regardless the way of reading the matrix: in line or in column (*e.g.*,  $1\rightarrow 2$  and  $2\leftarrow 1$ ).

From this dependence matrix we deduce the corresponding Petri net diagram. All services are explicitly represented as transitions, including the added services 18 and 19 even if they do not correspond to services offered by crisis actors. Note that the Petri net focuses on the behavioral dimension of crisis actors: only services to be executed and their synchronization is deduced. It is then possible for crisis cells to simulate, validate and analyze the behavior of the CRP.

<sup>1</sup> <http://gind.mines-albi.fr/en/projet/genepi>

ServiceId	ServiceName	ServiceActor
1	Check rainfall forecast	CrisisCell
2	Prepare for dyke supervision	CrisisCell
3	Dyke supervision	DDT
4	Dyke state evaluation	DDT
5	Decision-making for evacuation	CrisisCell
6	Send decision report to interior ministry	Prefet
7	Dyke supervision continuation	DDT
8	Prepare evacuation nursing home	ARS
9	Proceed to evacuation	ARS
10	Report on evacuation	CrisisCell
11	Send evacuation report to interior ministry	Prefet
12	Inform CPZCR of A71 cut-off	CrisisCell
13	Define cut-off strategy for A71	CPZCR
14	Implement cut-off strategy for A71	Security forces
15	Report on cut-off strategy	CPZCR
16	Send cut-off report to interior ministry	CPZCR
17	Asses the situation	CrisisCell

Table 1. Services required on day 7

ServiceId1	Type	ServiceId2
2	require	1
8	require	1
12	require	1
2	cause	3
3	cause	4
4	cause	5
5	cause	6
4	cause	7
17	follow	6
17	follow	7
8	cause	9
9	cause	10
10	cause	11
17	follow	11
12	cause	13
13	cause	14
14	cause	15
15	cause	16
17	follow	16

Table 2. Relations between services

ServiceId	useCondition
5	dykeState='low'
7	dykeSupervision='yes'

Table 3. Use conditions for services

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	■	→						→				→							
2	←	■	→																
3		←	■	→															
4			←	■	→													→	
5				←	■	→												#	
6					←	■											→		
7							■										→	←	#
8	←							■	→										
9								←	■	→									
10									←	■	→								
11									←	■								→	
12	←											■	→						
13												←	■	→					
14													←	■	→				
15														←	■	→			
16															←	■	→		
17						←	←				←						←	■	←
18				←	#		→											■	→
19							#										→	←	■

Table 4. Dependence matrix on day 7

Then the corresponding BPMN diagram is mapped from the Petri net. In the mapping process we label outgoing sequence flows from exclusive gateways with conditions that have to be checked. For instance, as illustrated in Fig. 5, the outgoing sequence flow from alternative gateway to service Decision-making for evacuation of actor CrisisCell is labelled with the following condition defining when the sequence flow is executed: dykeState='low'. Moreover, in the BPMN diagram, some services are labelled with risks while others are not. Regarding labelled services, this means that the risk is effectively taken into account when all the corresponding services have been performed. For instance, the risk of dyke failure in Saint-Privé Saint-

Mesmin will be taken into account, and thus will be removed from Risk/Damage in the meta-model, when the services Prepare for dyke supervision, Dykesupervision and Dyke state evaluation will be completed. Regarding non labelled services, they correspond to decision-making services (*e.g.*, Decision-making for evacuation), which rather generate new risks, or to services implementing hierarchical communication towards different ministries to which crisis cells are accountable (*e.g.*, Send decision report). Finally, as explained in Sec. 4, we also simplify the CRP removing services Start and End along with syntactic added services (*e.g.*, services 18 and 19). The mapping result is given in Fig. 4.



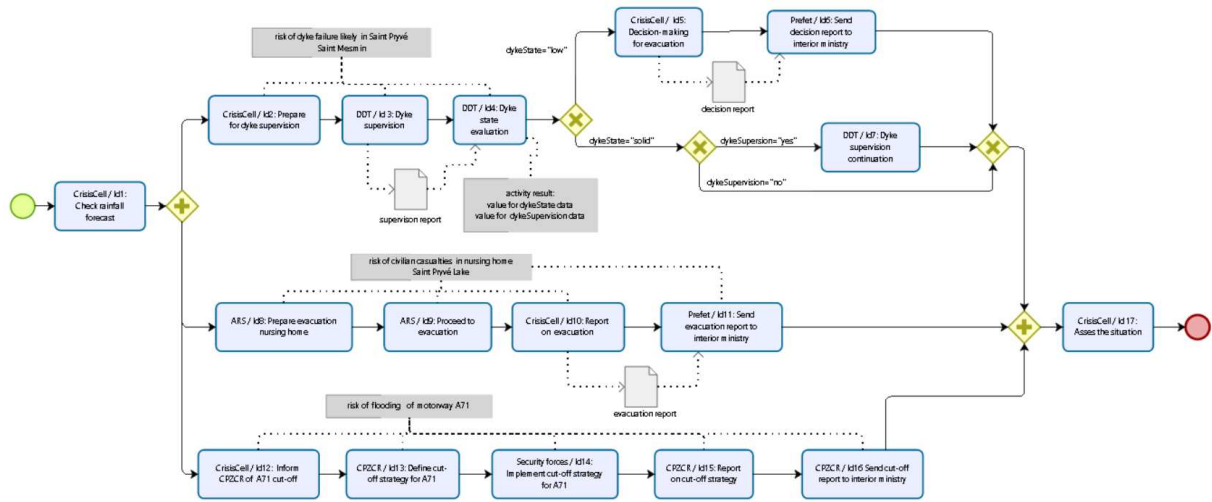


Fig. 4. Deduced CRP as a BPMN-like diagram

## 6. Discussion and conclusion

This paper has addressed deduction of processes driving crisis resolution from relevant facts observed in the field and business knowledge of actors involved in crisis resolution. Deduction step is a key step in crisis management (and in *GéNéPi*) as it provides crisis cells with guidelines for crisis resolution. Moreover these guidelines are consistent with both facts observed and crisis actors' knowledge. The recommended solution includes (i) a meta-model supporting facts and knowledge modelling and (ii) a set of algorithms implementing crisis resolution process deduction. In addition, the paper has reported on an experiment conducted in collaboration with a crisis cell, considering the flood of the Loire in June 2016 as a case study. The experiment took place in the context of the *GéNéPi* project, funded by the French research national agency. The experiment enabled us to assess difficulties of crisis cell members in modelling knowledge as instance of our meta-model. Fact modelling was really easy while knowledge modelling was more touchy, mainly the modelling of services and relations between them. Regarding services, the issue was their identification. Indeed, the *Préfet*, head of crisis cell, rather needs a macroscopic vision of the actions executed in the field, while the representatives of the different actors acting in the field need a more microscopic view, in which their actions are detailed. In the experiment, after discussion among the crisis cell members, we have modelled services according to *Préfet*'s needs. Regarding relations between services, require and cause relation were easily identified by crisis cells members themselves, but it took time to identify choice relations. In addition, the experiment enabled us to assess the match between the deduced

CRPs and what crisis cell members were expecting. Feedbacks were very good, as for each crisis day, the deduced CRP has matched with what the crisis cell would do, taking into account every risk or damage and not forgetting even one thing. Above all we have heard from crisis cell members that the inclusion of choices and decision-making services matched the way in which a crisis cell works, which is strength of our solution.

On the other side, the recommended deduction algorithm is a step forward with respect to existing solutions. Regarding existing deduction algorithms, unlike [5, 6], we recommend a declarative approach for knowledge modelling, thus making knowledge management easier to crisis cell members. Moreover we fully exploit this knowledge: (i) we deduce choices along with corresponding conditions, in addition to sequencing and parallelization of services and (ii) we label services with corresponding risk or damage. Labelling services is very important as it explains service selection in the CRP (thus the *Préfet* can defend undertaken actions to hierarchical authorities to which he is accountable) and it makes possible the deletion of risk or damage in the meta model (as they have been addressed in the field). Regarding process mining algorithms, our recommended algorithm extends the Alpha algorithm [8] building processes possibly starting with parallelism or alternative. Moreover, our algorithm does not need as input any log file as it only exploits knowledge for crisis actors for CRP deduction from facts observed.

However, three main improvements are required in our work. The first one is related to the consistency of modelled knowledge, and more precisely the consistency of relations between services. We did not investigate this point and have planned to do it shortly.

The second one is related to the filtering of services before Petri net deduction. So far, this filtering boils down to a unique selection of each service, even if it is required two times in the CRP. We will also address this key point in the next future. Finally, we are also interested in mining the social dimension that exists between actors involved in crisis cell resolution to determine in what this dimension influences crisis coordination.

## 7. References

- [1] O. Tahir, E. Andonoff, C. Hanachi, C. Sibertin-Blanc, F. Bénaben, V. Chapurlat, T. Lambolais, "A collaborative information system architecture for process-based crisis management", Int. Conference on Knowledge-Based Intelligent Information and Engineering System, Zagreb, Croatia, September 2008, pp. 630–641.
- [2] E. Andonoff, C. Hanachi, T. Le Nguyen Tuan, C. Sibertin-Blanc, "Interaction protocols for human-driven crisis resolution processes", IFIP working Conference on Virtual Enterprises, Albi, France, October 2015, pp. 63–76.
- [3] H. Ariouat, E. Andonoff, C. Hanachi, "Do process-based systems support emergent, collaborative and flexible processes? Comparative analysis of current systems", Int. Conference on Knowledge-Based Intelligent Information and Engineering System, York, United Kingdom, September 2016, pp. 511–520.
- [4] M. Dumas, M. La Rosa, J. Mendling, H. Reijers, "Fundamentals of Business Process Management", Springer, 2013.
- [5] F. Bénaben, N. Boissel-Dallier, H. Pingaud, JP. Lorré, "Semantic issues in model-driven management of information system interoperability", Int. Journal on Computer Integrated Manufacturing, 26(11), 2013, pp. 1042–1053.
- [6] G. Macé Ramète, M. Luras, L. Steffan, J. Lamothe, F. Bénaben, AM. Barthe-Delanoë, H. Dolidon, L. Lilas, "Towards a predictive model for decision support in road crisis management", Int. Conference on Digital Ecosystems and Technologies, Menlo Park, CA, USA, July 2013, pp. 137–140.
- [7] W. van der Aalst, "Process Mining – Data Science in Action", Springer, 2016.
- [8] W. van der Aalst, J. Weijters, L. Maruster, "Workflow Mining: Discovering process models from event logs", Int. Journal on Transactions on Knowledge and Data Engineering 16(9), 2004, pp. 1128–1142.
- [9] H. Schonenberg, B. Weber, B. van Dongen, W. van der Aalst, "Supporting flexible process through recommendations based on history". Int. Conference on Business Process Management, Milan, Italy, September 2008, pp. 51–66.
- [10] J. Ribeiro, J. Carmona, M. Misir, M. Sebag, "A recommender system for process discovery". Int. Conference on Business Process Management, Eindhoven, September 2014, The Netherlands, pp. 67–83.
- [11] M. Rangih, M. Comuzzi, B. Karakostas, "A framework to capture and reuse process knowledge in business process design and execution using social tagging". Int. Journal on Business Process Management, 22(4), 2016, pp. 835–859.
- [12] F. Bénaben, M. Luras, S. Truptil, N. Salatgé, "A meta-model for knowledge management in crisis management". Hawaiian Int. Conference on System Sciences, Koloa, HI, USA, January 2016, pp. 126–135.
- [13] F. Bénaben, C. Hanachi, M. Luras, P. Couget, V. Chapurlat, "A meta-model and its ontology to guide crisis characterization and its collaborative management". Int. Conference on Information System for Crisis Response Management, Washington, DC, USA, May 2008, pp. 189–196.
- [14] W. van der Aalst, "The application of Petri Nets to Workflow Management", Int. Journal on Circuits, Systems and Computers 8(1), 1998, pp. 21–66.
- [15] A. de Medeiros., B. van Dongen, W. van der Aalst, A. Weijters, "Process mining: extending the  $\alpha$ -algorithm to mine short loops", Eindhoven Technology University, 2004, <https://pure.tue.nl/ws/files/1864325/576199.pdf>
- [16] L. Wen, W. van der Aalst, J. Wang, J. Sun, "Mining process models with non-free-choice constructs", Int. Journal on Data Mining and Knowledge Discovery 15(2), 2007, pp. 145–180.
- [17] ProM: Process mining toolkit. <http://www.promtools.org/doku.php>